#SHAREorg

SHARE
Technology · Connections · Results

Introduction to CA AppLogic® for System z

Summer Spaulding
CA Technologies, Inc.

August 13, 2013
Session 13485

SHARE in Boston

Abstract

AppLogic allows web applications to be deployed virtually on a scalable grid using a drag-and-drop web browser interface. Applications can scale from a fraction of a server up to the whole grid, based on current demand. CA AppLogic for System z extends this support to Linux guests under z/VM, allowing your mainframe to participate as a resource in the computing grid and play a key role in your private cloud infrastructure.

In this session, the speaker will present a technical overview and demo of CA AppLogic for System z.

Agenda

- Current Deployment Options for Linux on System z
- CA AppLogic for System z - The Grid Architecture
- Closer Look at the Solution
- Using CA AppLogic for System z
- Summary

Agenda

- **Current Deployment Options for Linux on System z**
- CA AppLogic for System z - The Grid Architecture
- Closer Look at the Solution
- Using CA AppLogic for System z
- Summary

Current Deployment Options

Just provision Linux container

Add virtual infrastructure and/or middleware

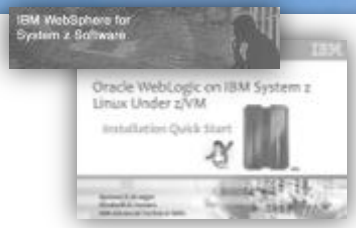
Provision entire application stack

Current Deployment Options

Just provision Linux container

Clone a golden image

Add virtual infrastructure and/or middleware



Script customization

Provision entire application stack

Manually install and configure

Evaluating Deployment Options

Provisioning just Linux or entire application? Consider…

Provisioning

- Speed to value
- Accuracy
- Auditing and reporting
- Resource allocation and constraints

Ongoing Management

- Manage to service level objectives
- Linux patching and upgrades
- Component patching and upgrades etc.
- Charge/show back

Controlled De-provisioning

- End of application lifecycle
- Varying capacity demands
- Efficient use of system resources

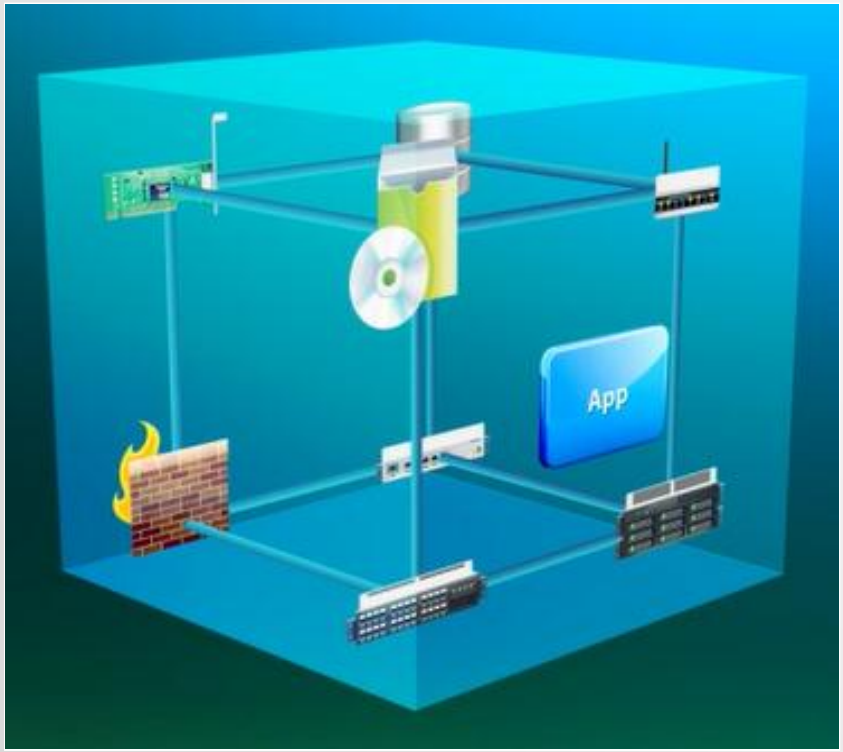
CA AppLogic® for System z
different deployment approach

Virtualize Linux on System z application and its ENTIRE infrastructure

- Firewalls
- Load balancers
- Web servers
- App servers
- Storage

CA AppLogic® for System z
Virtualized Business Service

Create, test, provision, deploy and manage it all as a single unit called a ***Virtualized Business Service***



Virtualized Business Service

benefits of a virtual business service

ability to deploy applications & services in minutes

More Agility for Enterprises

- Build and deploy apps using appliances
- On-demand elasticity and flexibility
  - Migrate entire apps instantly
  - Replicate and scale apps instantly
- Work through an intuitive GUI, not by pulling cables and copying gold images

Leverage Power of Linux on System z

- Increase RASSS
- Reduce datacenter costs
- Reduce management costs
- Easier interoperability with z/OS
- Power of cloud platform

Agenda

- Current Deployment Options for Linux on System z
- **CA AppLogic for System z - The Grid Architecture**
- Closer Look at the Solution
- Using CA AppLogic for System z
- Summary

What is CA AppLogic® for System z?

Grid operating system that runs and scales existing Linux-based applications as Linux on System z guests on a z/VM system

Capabilities Include:

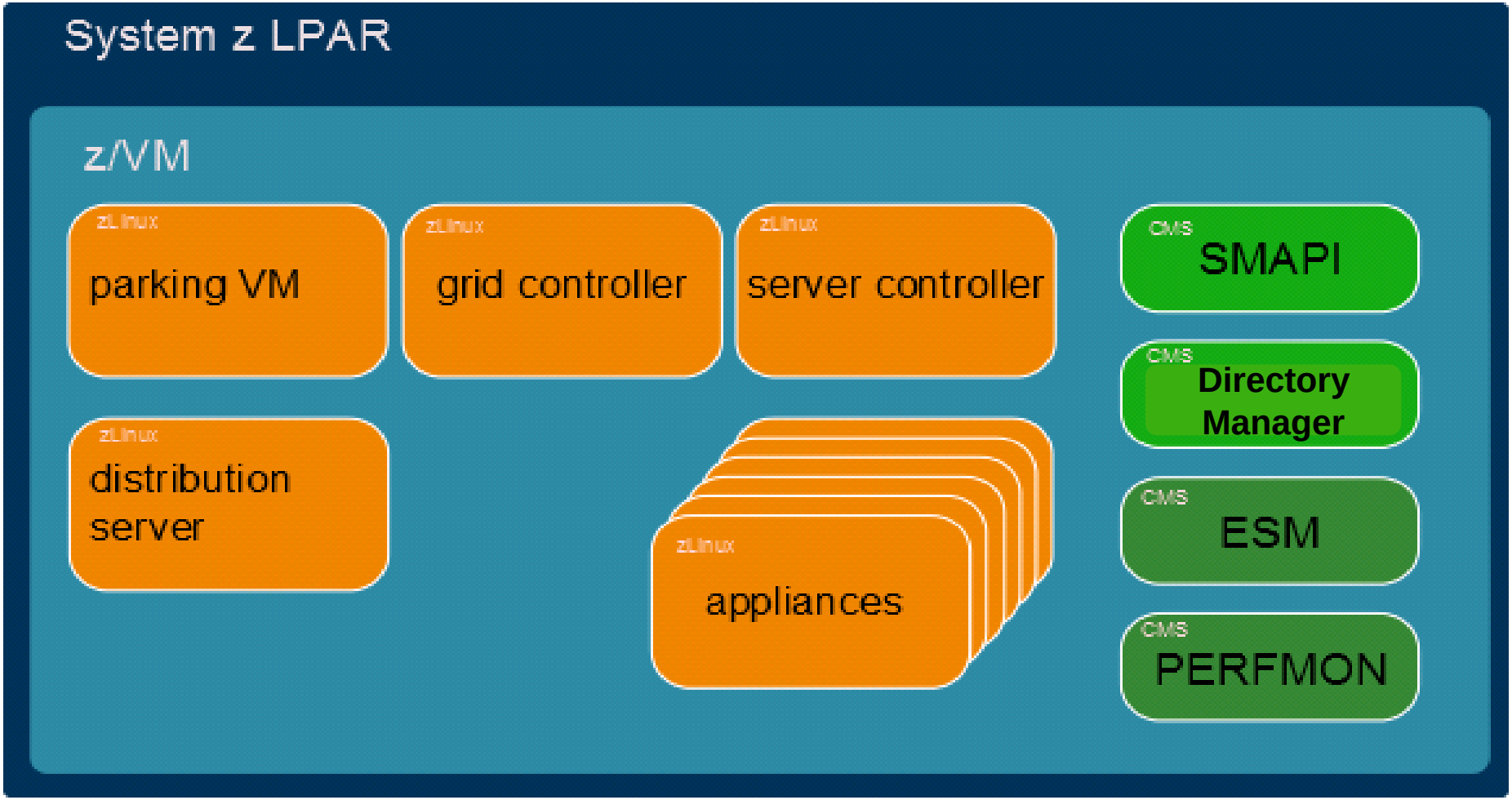
- Deploy existing Linux-based applications on a grid without changing any code
- Scale the resources used by each application from a fraction of a server up to the whole server
- Manage all applications and storage with only a web browser
- Define role-based access levels for grid and grid resources (i.e., applications)
- Create and maintain standard virtual server builds as appliances – presented through GRID catalog
- Create cataloged z/OS resource gateway appliances for use in applications:
  - Datacom
  - IDMS
  - CICS
  - DB2
  - IMS

Vendor-neutral and supports open source middleware such as Apache, MySQL, and JBoss

Grid Architecture
Basic Premises

- **AppLogic grid will run under z/VM on a System z LPAR**
- **Grid cannot span multiple LPARs**
- **LPAR and the z/VM will run only AppLogic**
  - Exception of service applications such as directory management, external security manager, performance management, SMAPI
- **There will be only one grid running on the LPAR**
- **Current Linux distributions certified AppLogic for System z**
  - Red Hat Enterprise Linux 6.x
  - CentOS 6.x
  - SUSE Linux Enterprise Server 11 SP 2
- **Appliance Kit (APK) required on each virtual machine running on grid**
  - Set of scripts that enable communications with the grid controller, auto-configuration of the appliance network interfaces, and the ability to obtain property values from the appliance boundary

Grid Architecture



AppLogic Element

Required System Element

Optional System Element

Grid Architecture

Architecture Element	Description
Distribution Server (AL)	• Linux guest where AppLogic® for System z software is installed • Used to create and deploy the grid controller, server controller, and appliances using the command line tool, ALDO • Stateless: Any instance can perform upgrades and changes to a grid to which it has network access
Appliance	• Building blocks for applications • Through a user interface, user can create applications using cataloged appliances; which can be scaled and provisioned as required
SMAPI	• Provides Server Controller's interface to Directory Manager
Directory Manager	• Provides interface to the user directory where appliances are created on demand • Supports CA VM:DIRECT, CA VM:SECURE, and IBM DIRMAINT
ESM (External Security Manager)	• Manages permissions for creation of users and disk access • Supports CA VM:SECURE and IBM RACF. (OPTIONAL)
IBM PERFMON	• Resource utilization monitoring. (OPTIONAL)

Parking Machine

- **Created with each GRID**
- **Owns the GRID Controller**
- **Owns all of the GRID disks**
- **As User entry in the directory**

```
USER ALVO0101 NOLOG    128K 1M G 64
* Parking machine for grid 01
MDISK 0201 3390 1 2500 LNX00F M
MDISK 0202 3390 1 1500 LNX00E M
MDISK 0203 3390 1 8000 LNX00D M
MDISK 0000 3390 1 835 LNX00C WR
MDISK 0001 3390 1 835 LNX00B WR
MDISK 0002 3390 2501 353 LNX00A WR
MDISK 0003 3390 2501 675 LNX00F WR
```

Example user directory entry

- **USER** defines basic properties (name, virtual memory size, privilege class)
- **MDISK** defines minidisks within GRID

Server Controller (SC)

- **Linux Virtual Machine created with each GRID**
- **Manages the provisioned applications**
- **Performs all interfacing with z/VM (virtual machine creation and deletion, minidisk creation and deletion, and so on), using SMAPI and vmcp**

Example user directory entry

- **USER** defines basic properties (name, virtual memory size, privilege class)
- **LOGONBY** establishes ALDO VM access
- **IUCV** establishes communications link to grid
- **SHARE** defines resource (e.g. CPU) usage
- **COMMAND** defines storage resources
- **NICDEF**s define virtual NICs
- **IPL & MDISK** define LINUX boot device

```
USER ALSC01 LBYONLY  1024M 2048M BCDEG 64
* Server Controller for grid 01
MACHINE ESA
LOGONBY ALDO
IUCV ALLOW
IUCV *VMEVENT
OPTION LNKNOPAS  LNKSTABL LNKEXCLU
SHARE ABSOLUTE 5% RELATIVE 100 LIMITSOFT
COMMAND DEFINE STORAGE AS 1024M STANDBY
1024M
IPL 0201
CONSOLE 0009 3215 T ALGC01
NICDEF C000 TYPE QDIO DEVICES 3
NICDEF C050 TYPE QDIO DEVICES 3
MDISK 0201 3390 1 2500 LNX00A M
```

Grid Controller (GC)

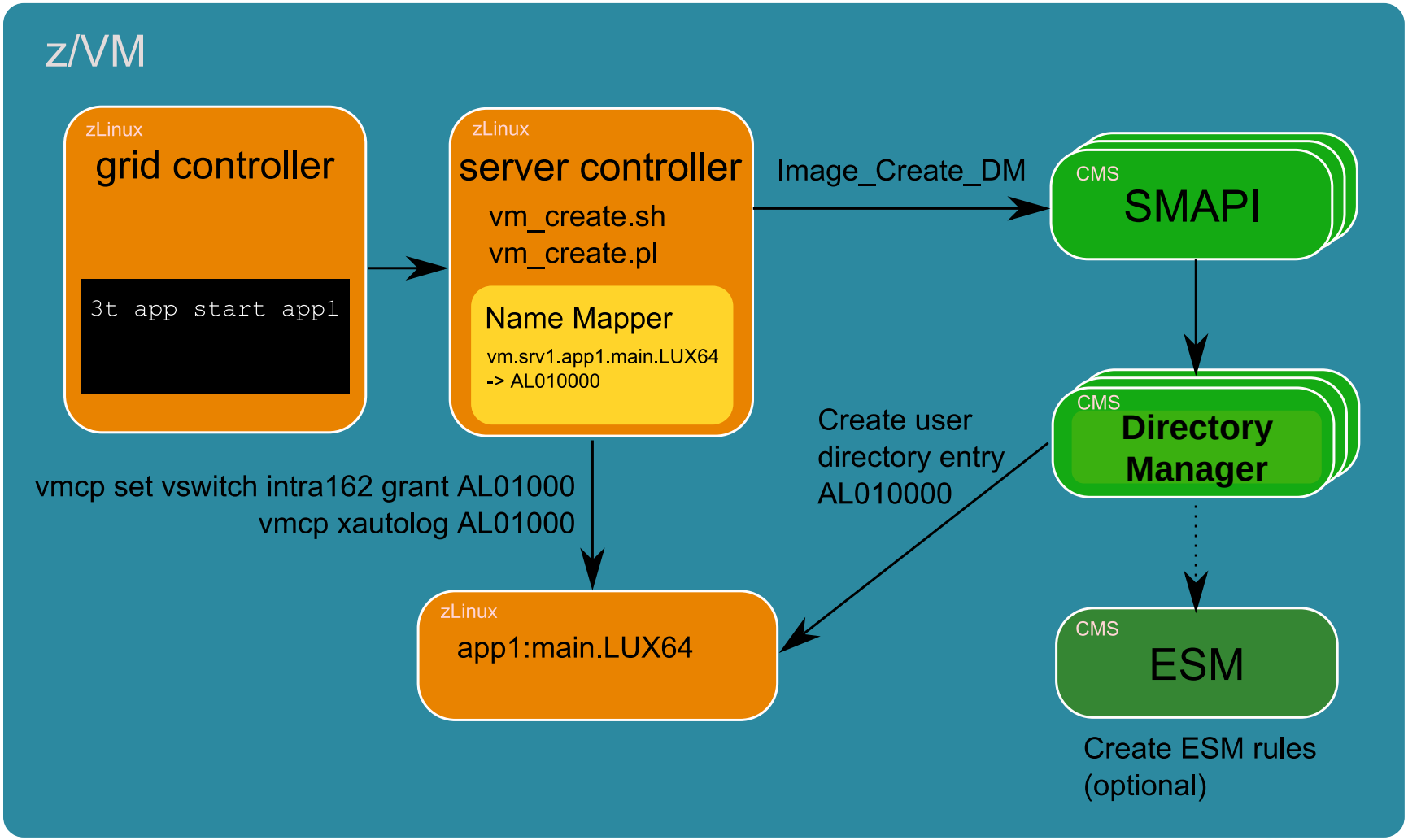
- **Linux Virtual Machine created with each GRID**
- **Manages and monitors the grid**
- **Serves the web user interface**
- **Handles the command line shell commands**

Example user directory entry

- **USER** defines basic properties (name, virtual memory size, privilege class)
- **LOGONBY** establishes ALDO VM access
- **IUCV** establishes communications link to grid
- **SHARE** defines resource (e.g. CPU) usage
- **COMMAND** defines storage resources
- **NICDEF**s define virtual NICs
- **LINK** defines links to minidisks owned by the parking machine
- **IPL** establishes LINUX boot device

```
USER ALGC01 LBYONLY  512M 2048M G 64
* Grid Controller for grid 01
MACHINE ESA
LOGONBY ALDO
IUCV ALLOW
OPTION LNKNOPAS  LNKSTABL LNKEXCLU
SHARE ABSOLUTE 5% RELATIVE 100 LIMITSOFT
COMMAND DEFINE STORAGE AS 1024M STANDBY 1024M
IPL 0201
CONSOLE 0009 3215
NICDEF C000 TYPE QDIO DEVICES 3
NICDEF C050 TYPE QDIO DEVICES 3
LINK ALVO0101 0201 0201 M
LINK ALVO0101 0202 0202 M
LINK ALVO0101 0203 0203 M
```

Grid *in action*



Appliance Virtual Machines

Appliance entries created on demand

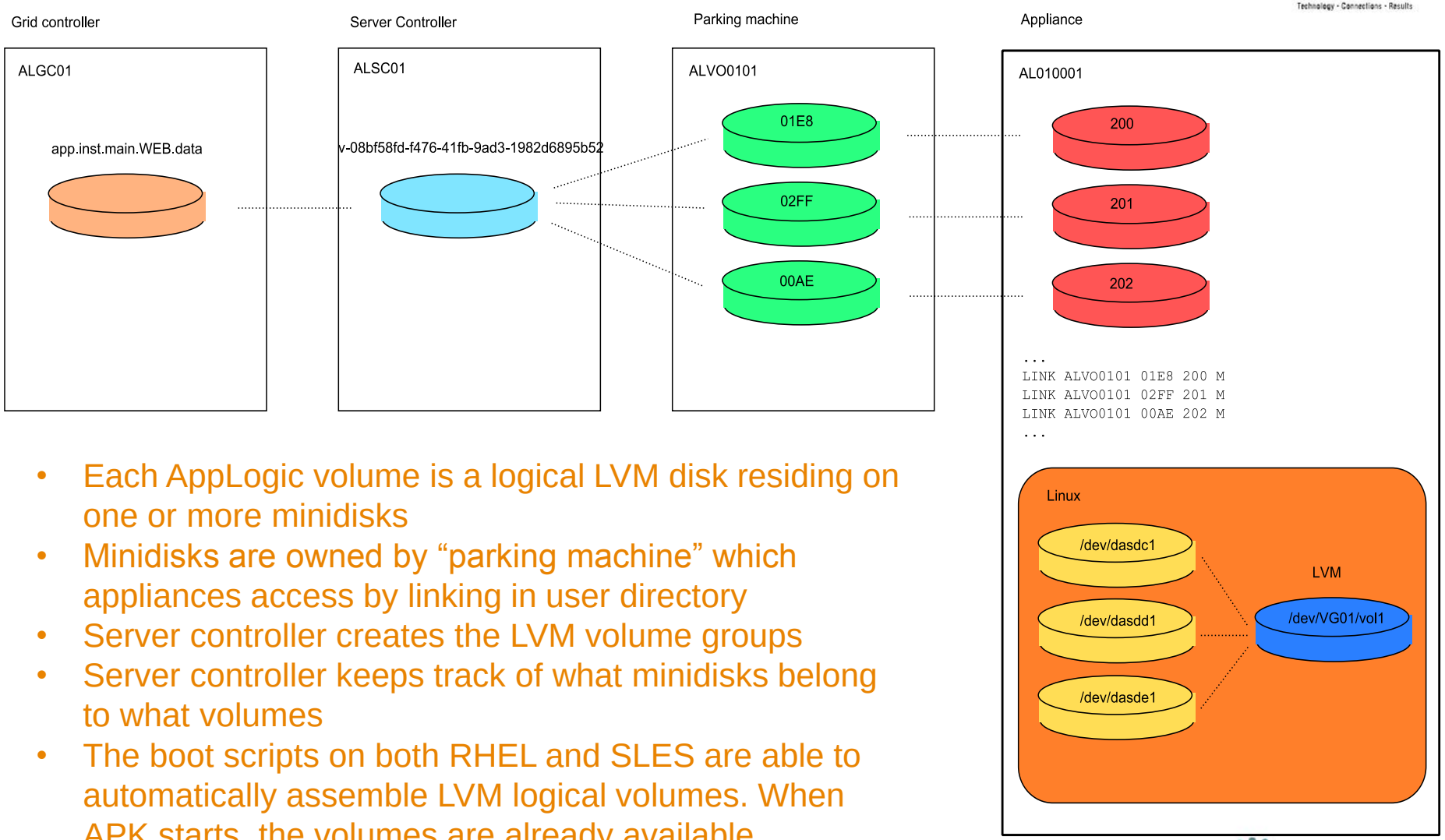
- As User entry in the directory
- Usually requiring permission of External Security Manager

Example user directory entry

- USER defines basic properties (name, virtual memory size, privilege class)
- SHARE defines resource (e.g. CPU) usage
- LINK defines links to minidisks owned by the parking machine
- NICDEFs define virtual NICs

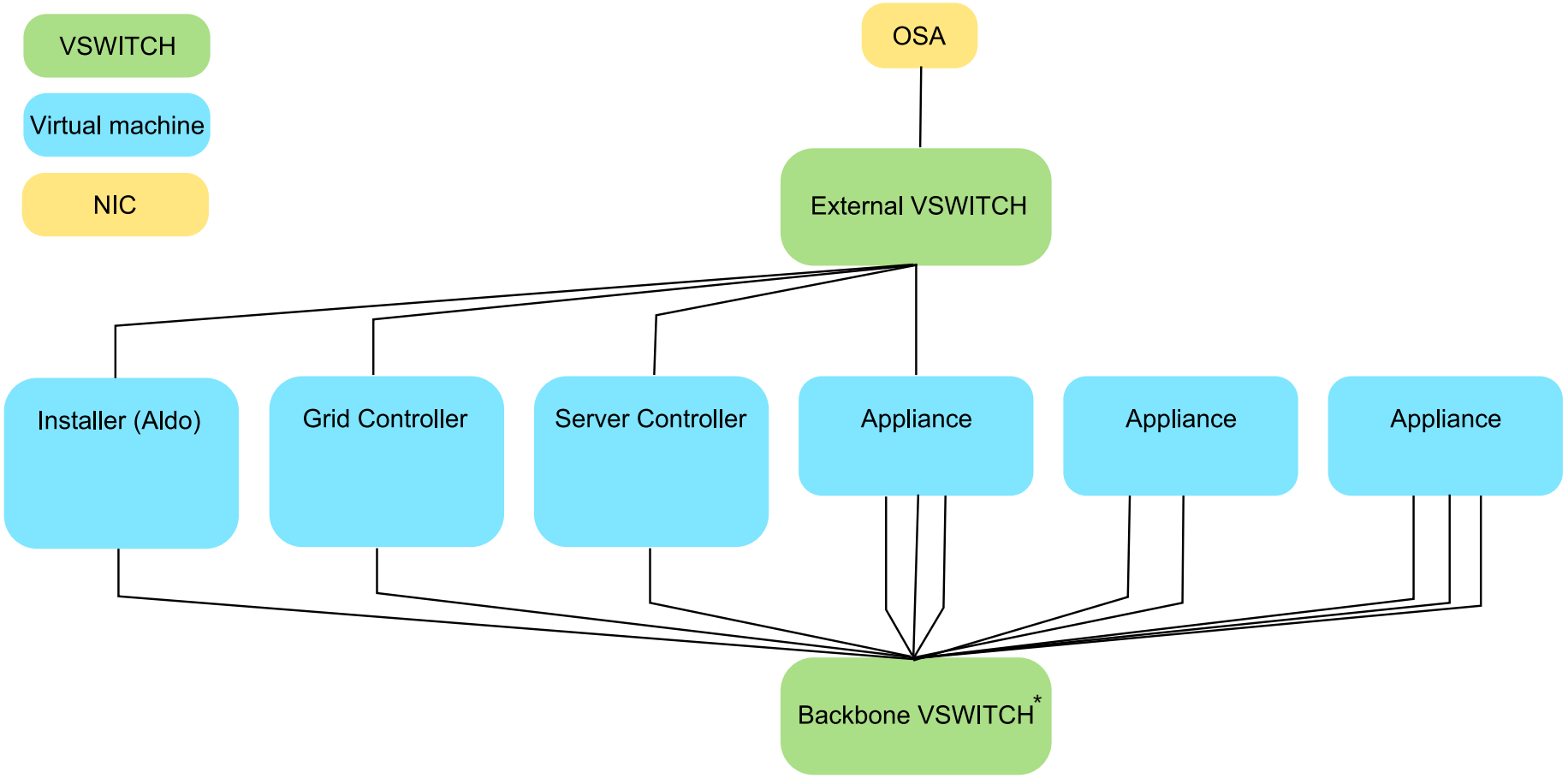
```
USER AL010000 XXXXXXXX 512M 512M G 64
MACHINE ESA
COMMAND SET RUN ON
SHARE RELATIVE 50 ABSOLUTE 2.5% LIMITSOFT
IUCV ALLOW
IPL 0203
XAUTOLOG ALSC01
CONSOLE   0009 3215 T ALSC01
LINK ALVO0201 0028 0203 M
NICDEF C000 TYPE QDIO LAN SYSTEM ALBL01
NICDEF C000 DEVICES 3 MACID 101C00
NICDEF C003 TYPE QDIO LAN SYSTEM ALBL01
NICDEF C003 DEVICES 3 MACID 101C01
NICDEF C006 TYPE QDIO LAN SYSTEM ALBL01
NICDEF C006 DEVICES 3 MACID 101C02
NICDEF C009 TYPE QDIO LAN SYSTEM ALBL01
NICDEF C009 DEVICES 3 MACID 101C03
NICDEF C00C TYPE QDIO LAN SYSTEM ALBL01
NICDEF C00C DEVICES 3 MACID 101C04
```

Disks



- Each AppLogic volume is a logical LVM disk residing on one or more minidisks
- Minidisks are owned by “parking machine” which appliances access by linking in user directory
- Server controller creates the LVM volume groups
- Server controller keeps track of what minidisks belong to what volumes
- The boot scripts on both RHEL and SLES are able to automatically assemble LVM logical volumes. When APK starts, the volumes are already available.

Networking



***No physical server-to-server communications needed – managed as "virtual wire" with non-routable IPs**

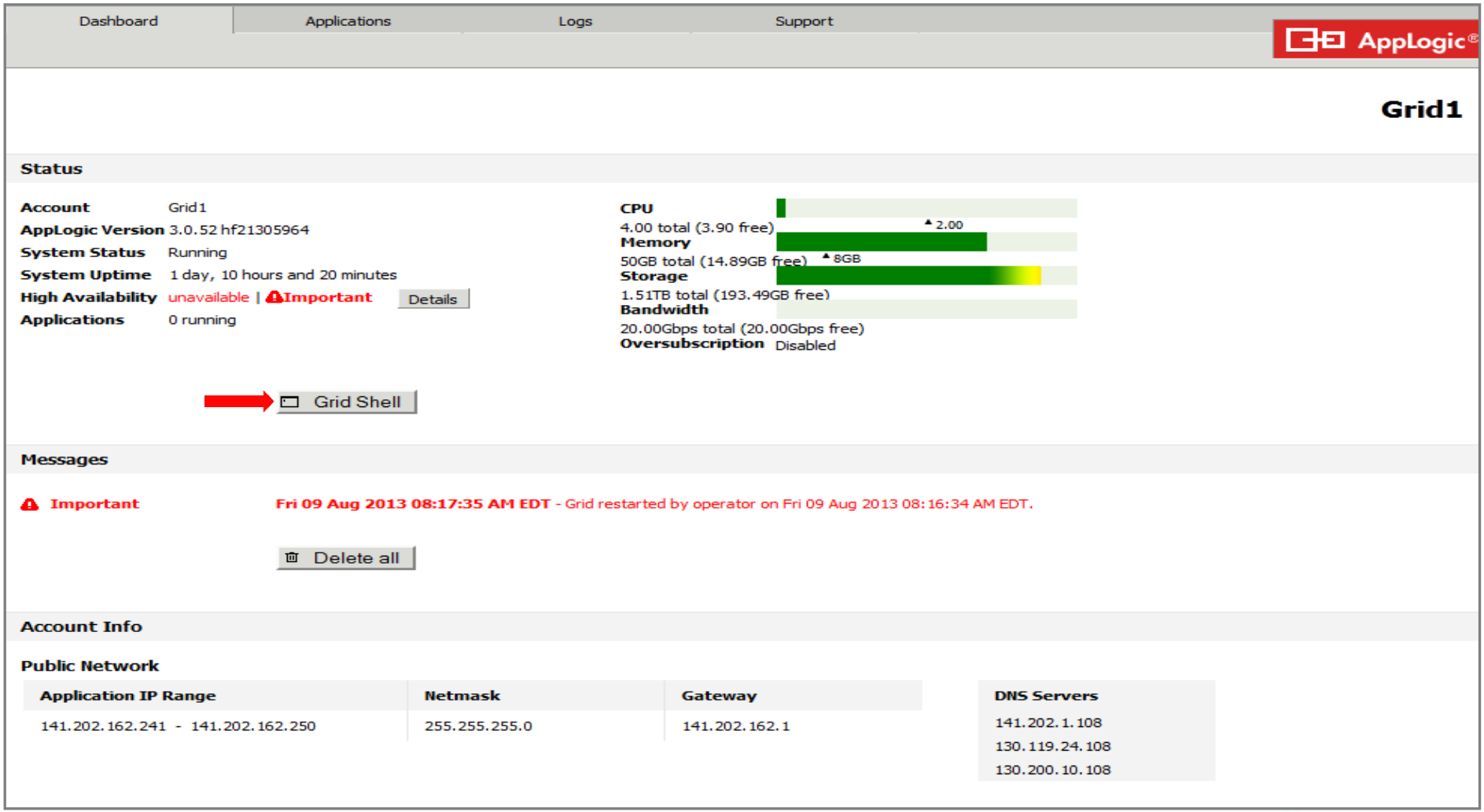
vm2class utility

- **Can take a zLinux guest running on the same z/VM as AppLogic, make a copy and convert it to an appliance.**
- **By default, installs the appropriate appliance kit (APK) and creates a managed appliance**
- **zLinux must be prepared for conversion – boot and package requirements**
- **After the zLinux guest is converted, it should be improved with boundary definition and configuration scripts to become a full-feature appliance – but will run as converted**
  - **Example:**
    3t util vm2class app_name=sles userid=slesgold vol1_addr=0203
    - **app_name –** Name of the newly created application
    - **userid –** z/VM name (userid) of the zLinux guest to be converted
    - **vol1_addr –** Virtual address of minidisk where zLinux data is located (can have vol2_addr etc. if the guest has multiple minidisks)

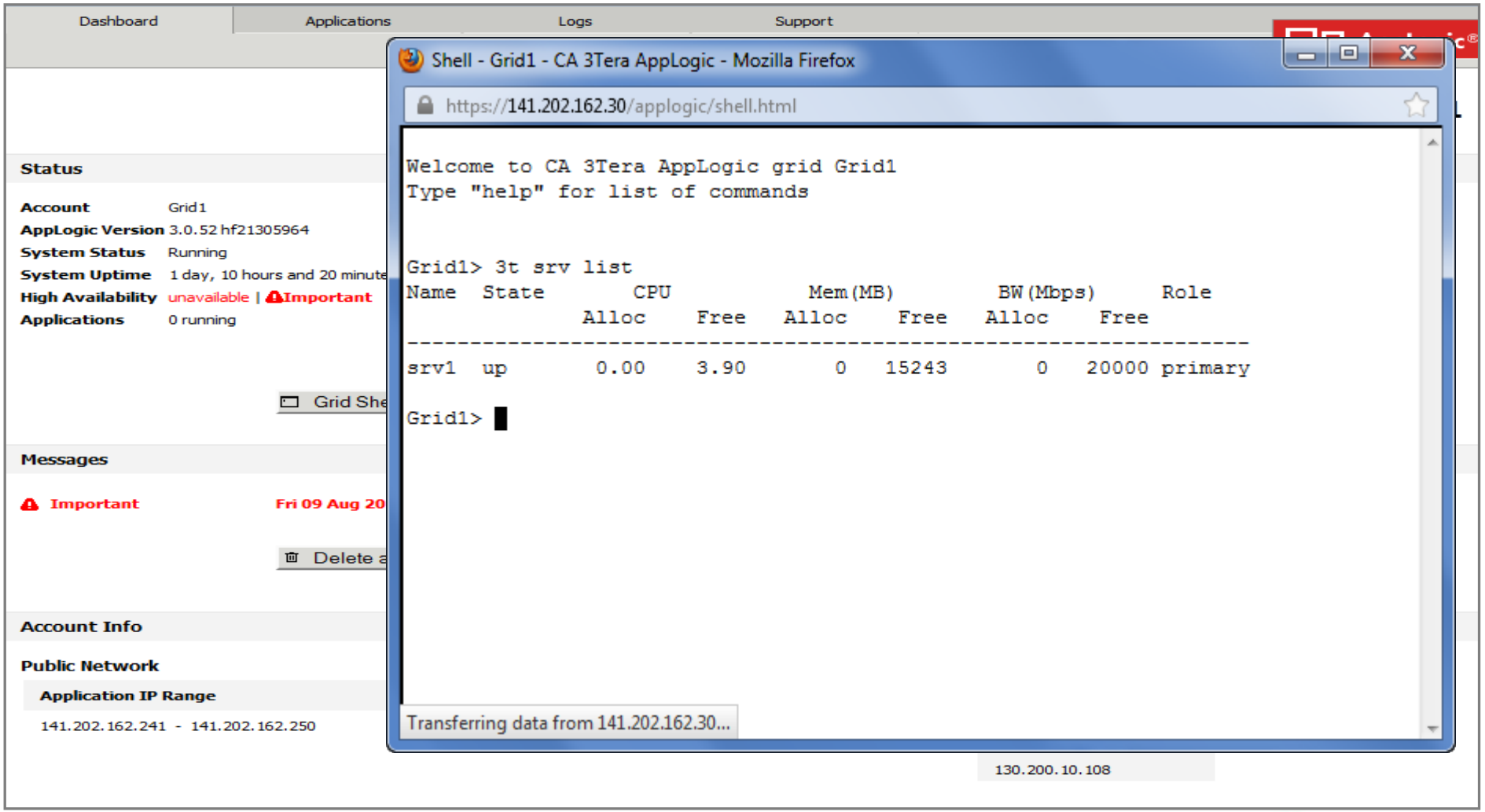
Agenda

- Current Deployment Options for Linux on System z
- CA AppLogic for System z - The Grid Architecture
- **Closer Look at the Solution**
- Using CA AppLogic for System z
- Summary

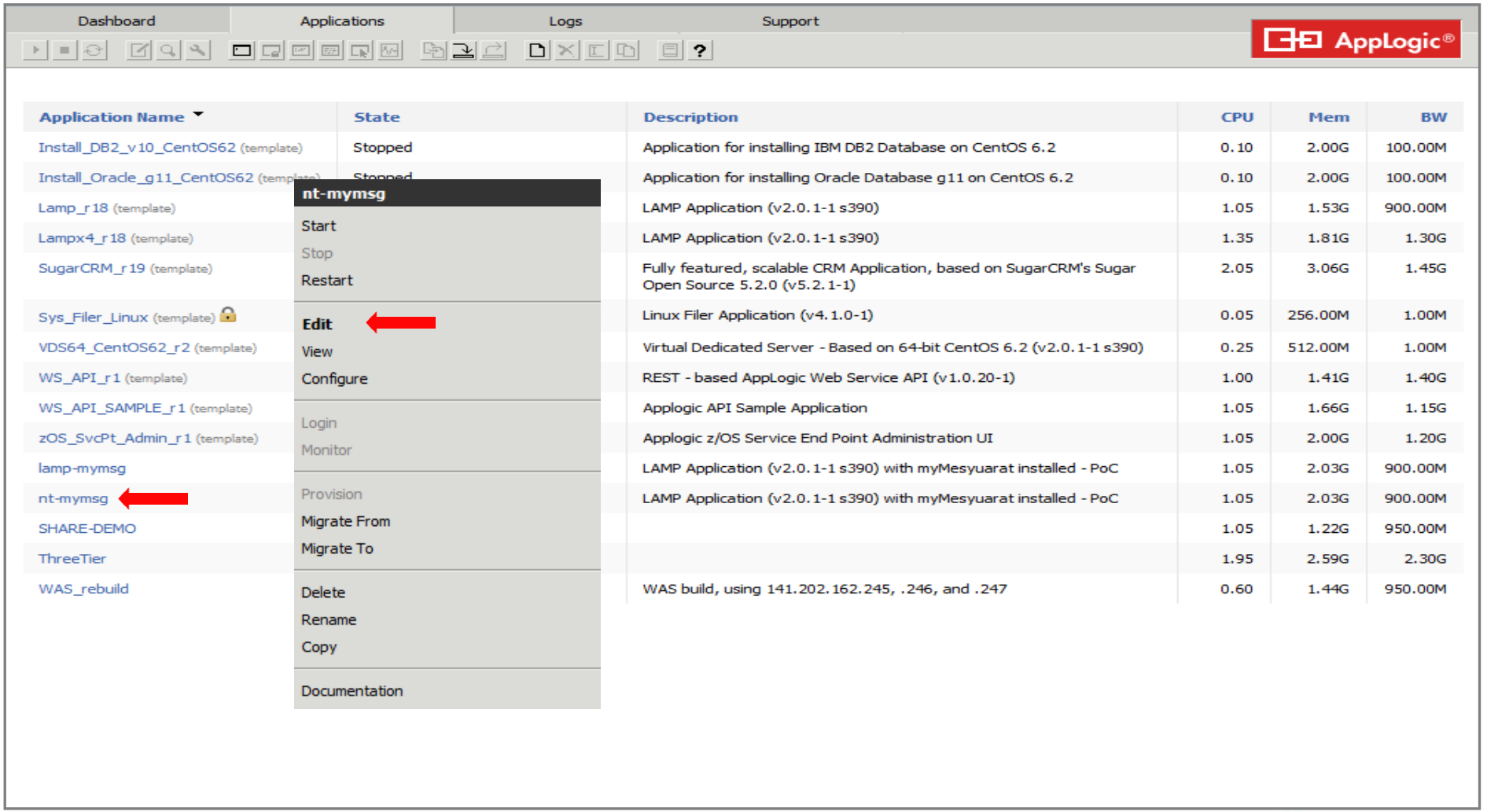
CA AppLogic® for System z
System Dashboard



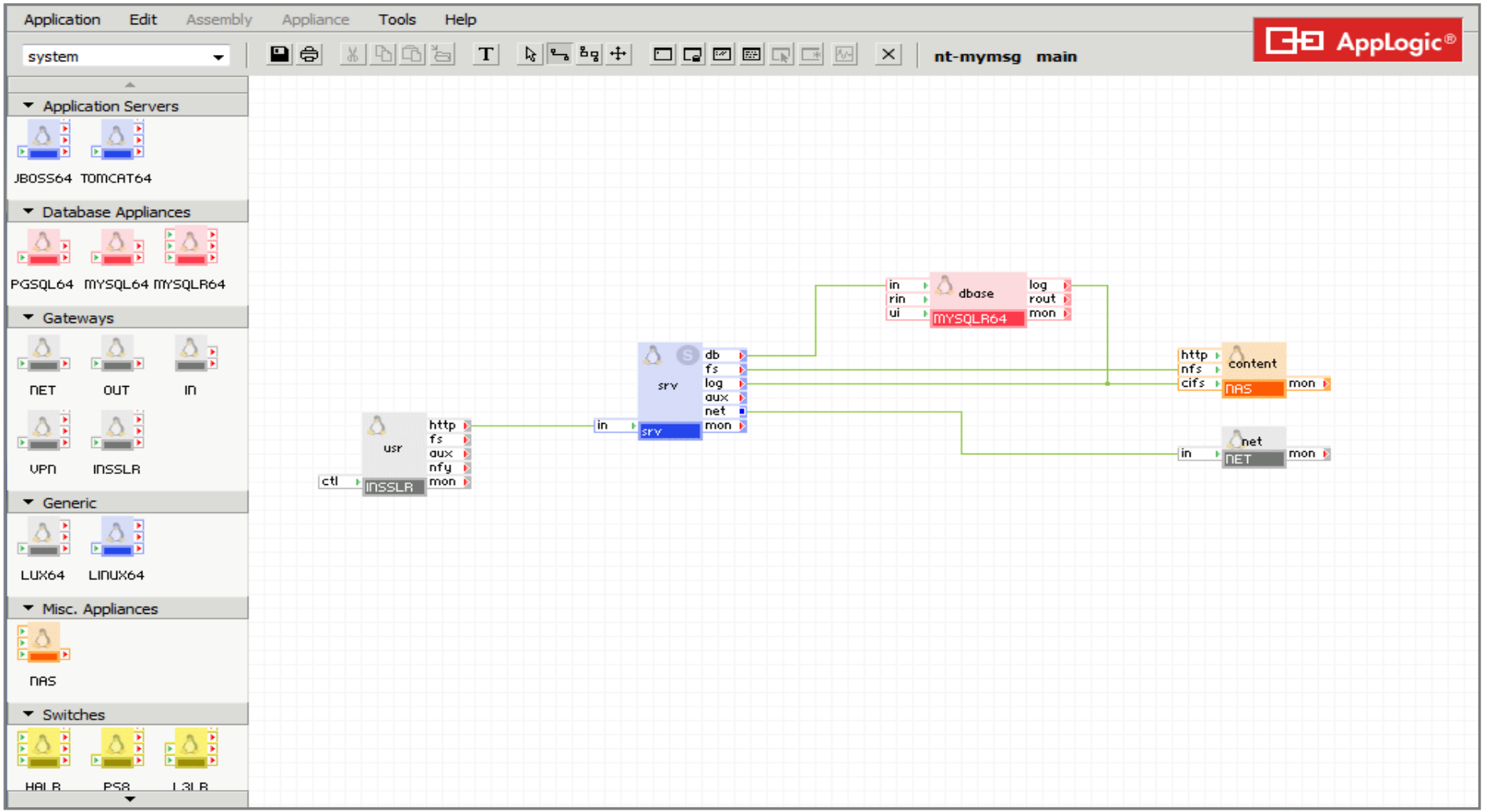
CA AppLogic® for System z
System Dashboard



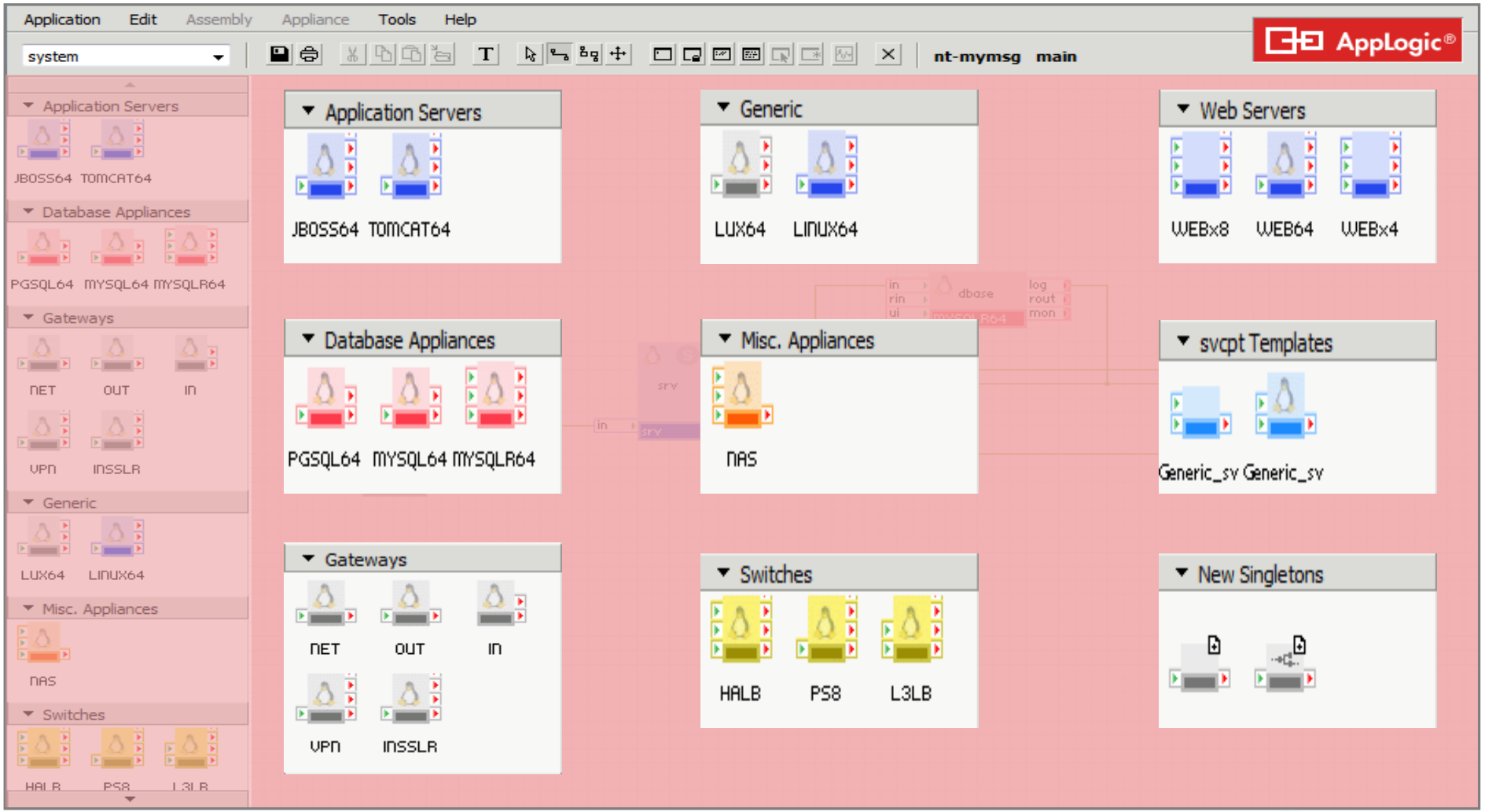
CA AppLogic® for System z
Applications



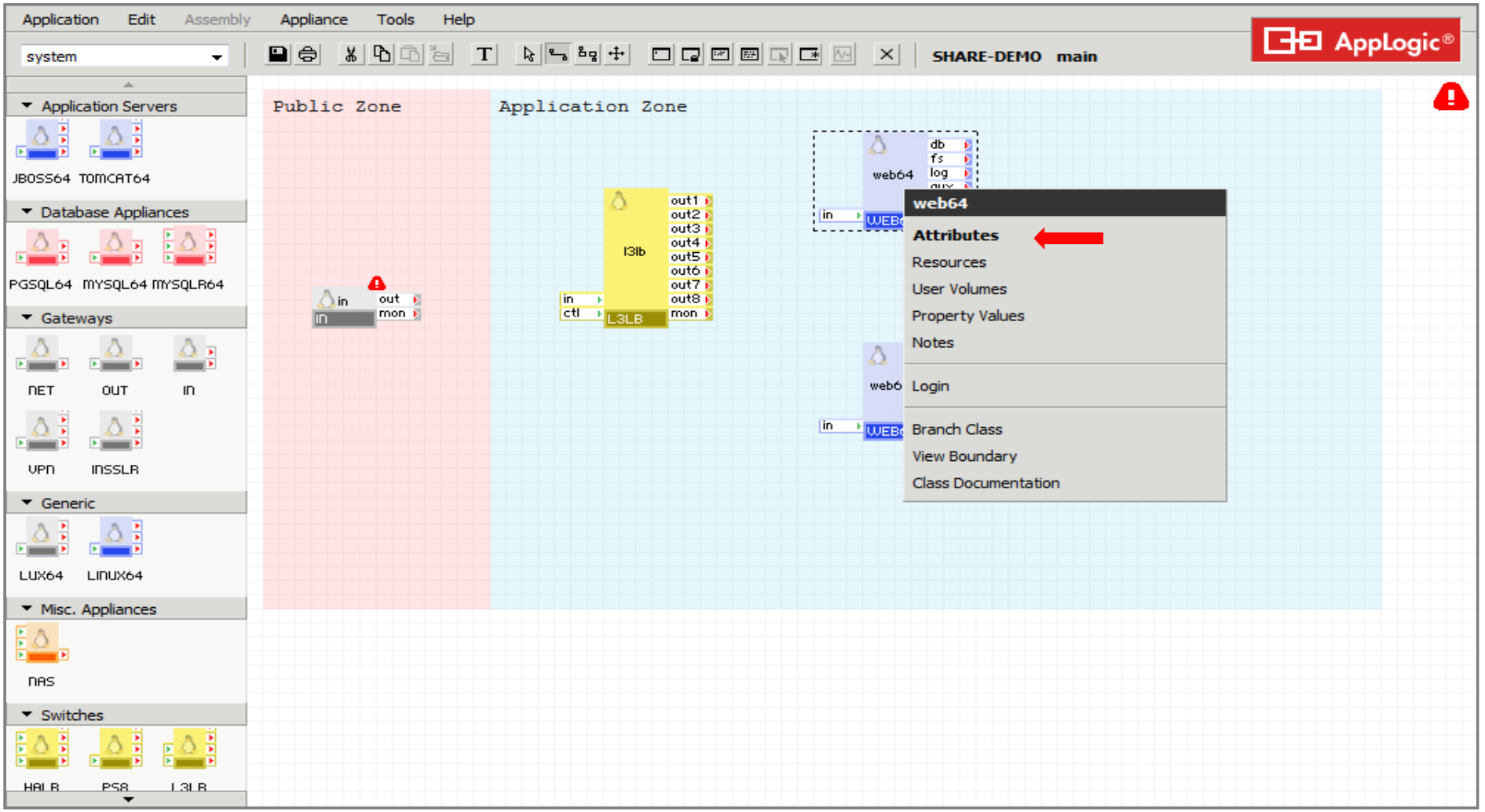
CA AppLogic® for System z
Infrastructure Editor



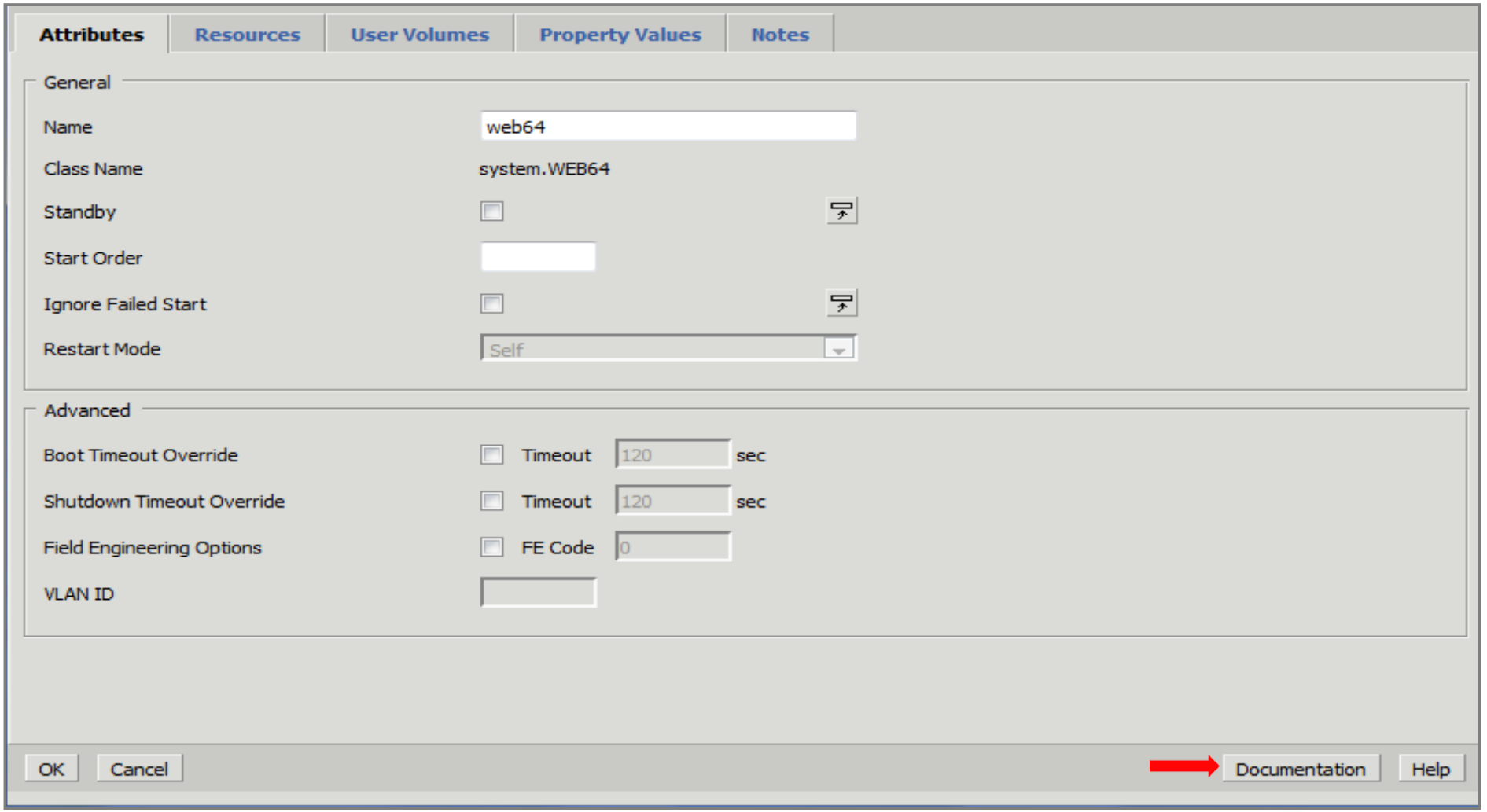
CA AppLogic® for System z
Infrastructure Editor – Catalog



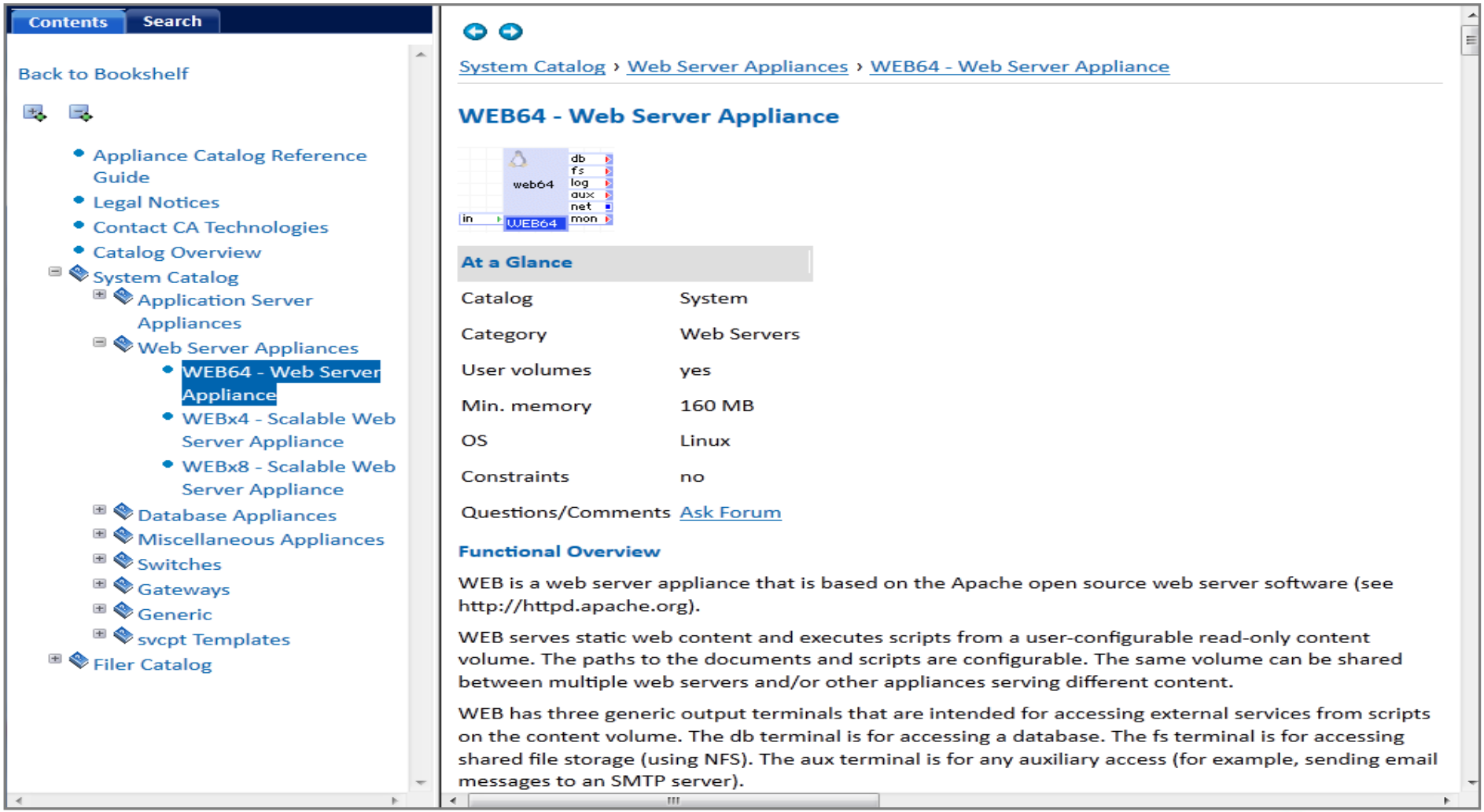
CA AppLogic® for System z
Infrastructure Editor – Appliance Instances



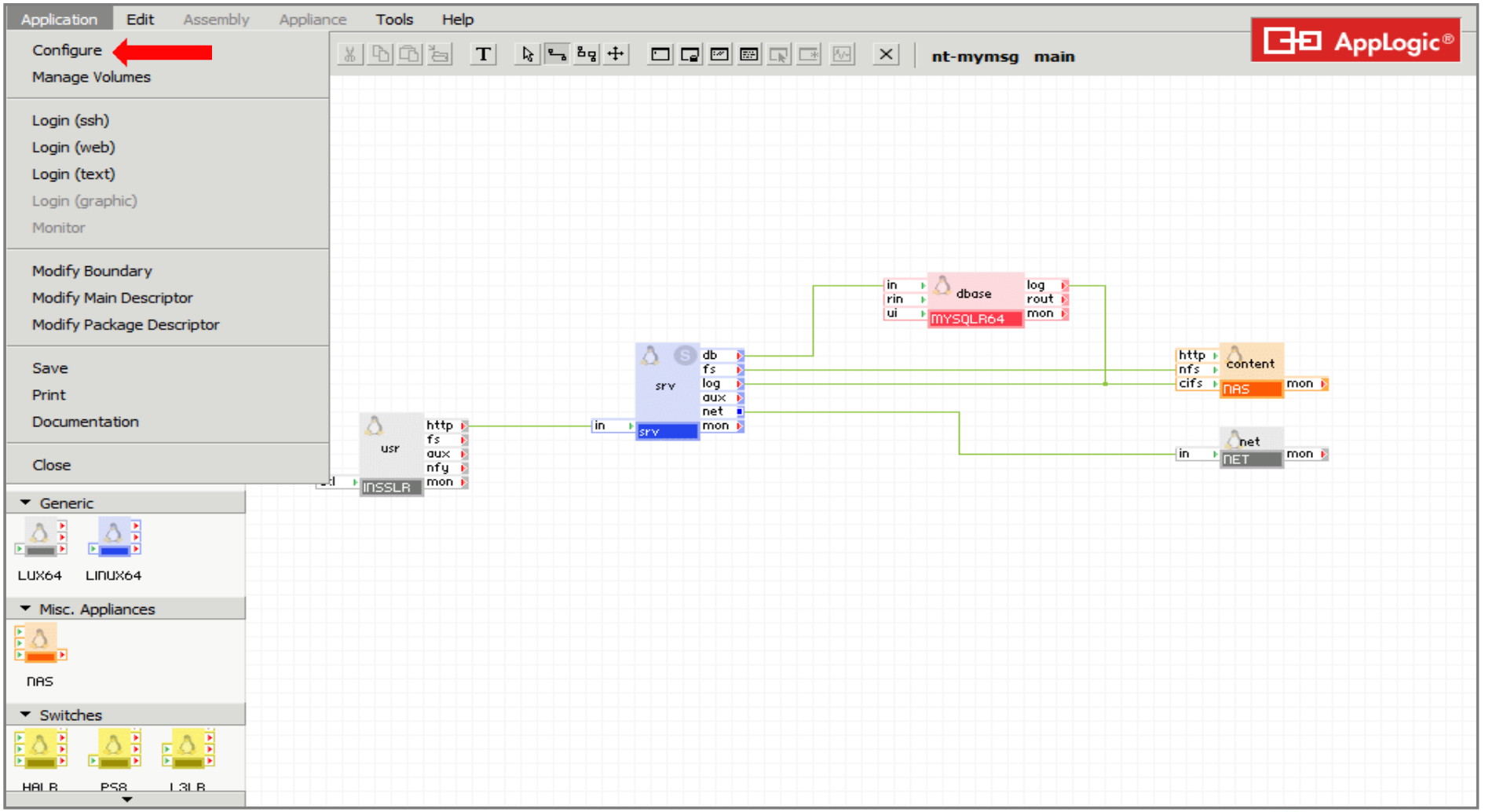
CA AppLogic® for System z
Infrastructure Editor – Appliance Instances



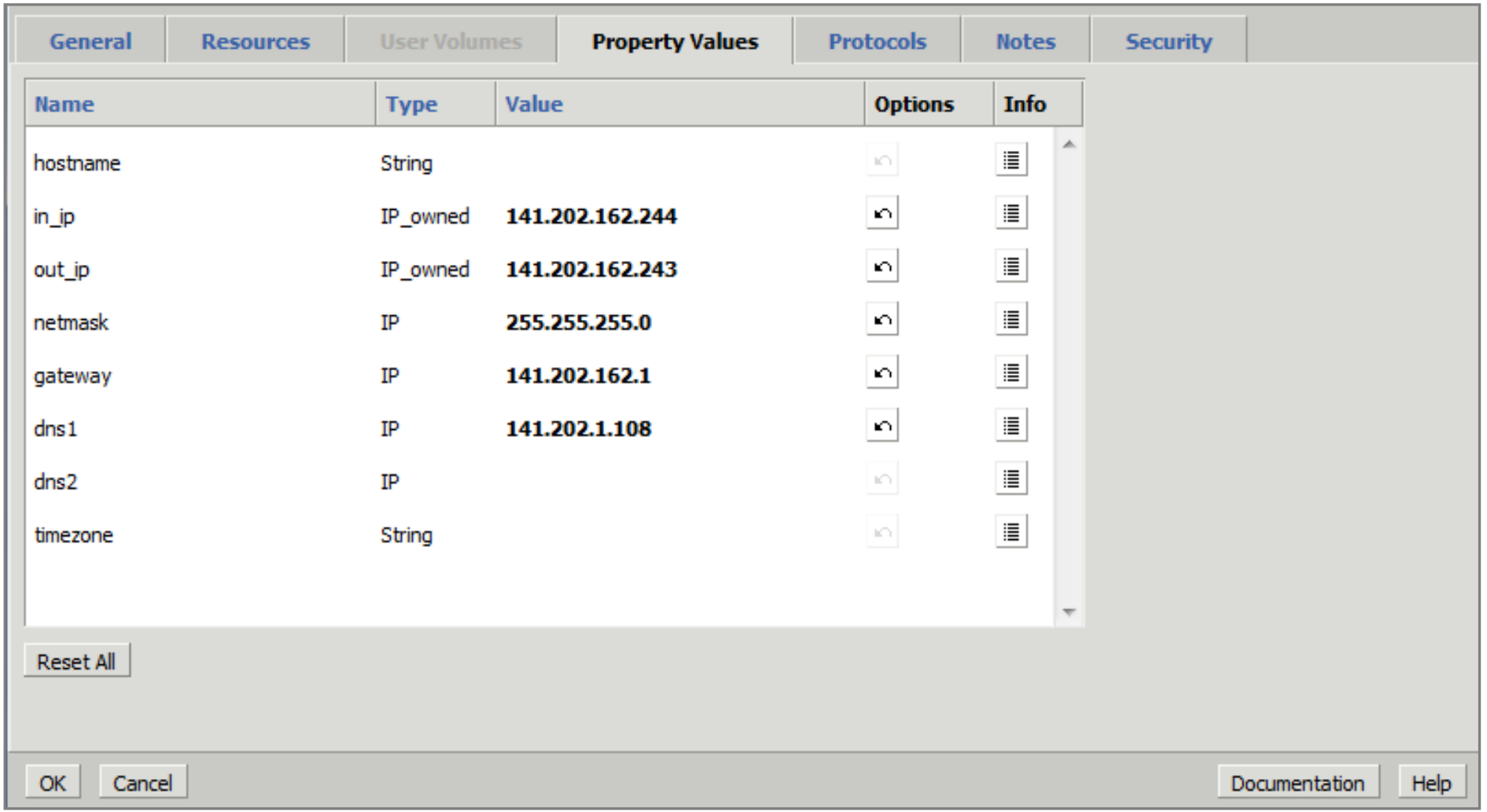
CA AppLogic® for System z
Infrastructure Editor – Appliance Instances



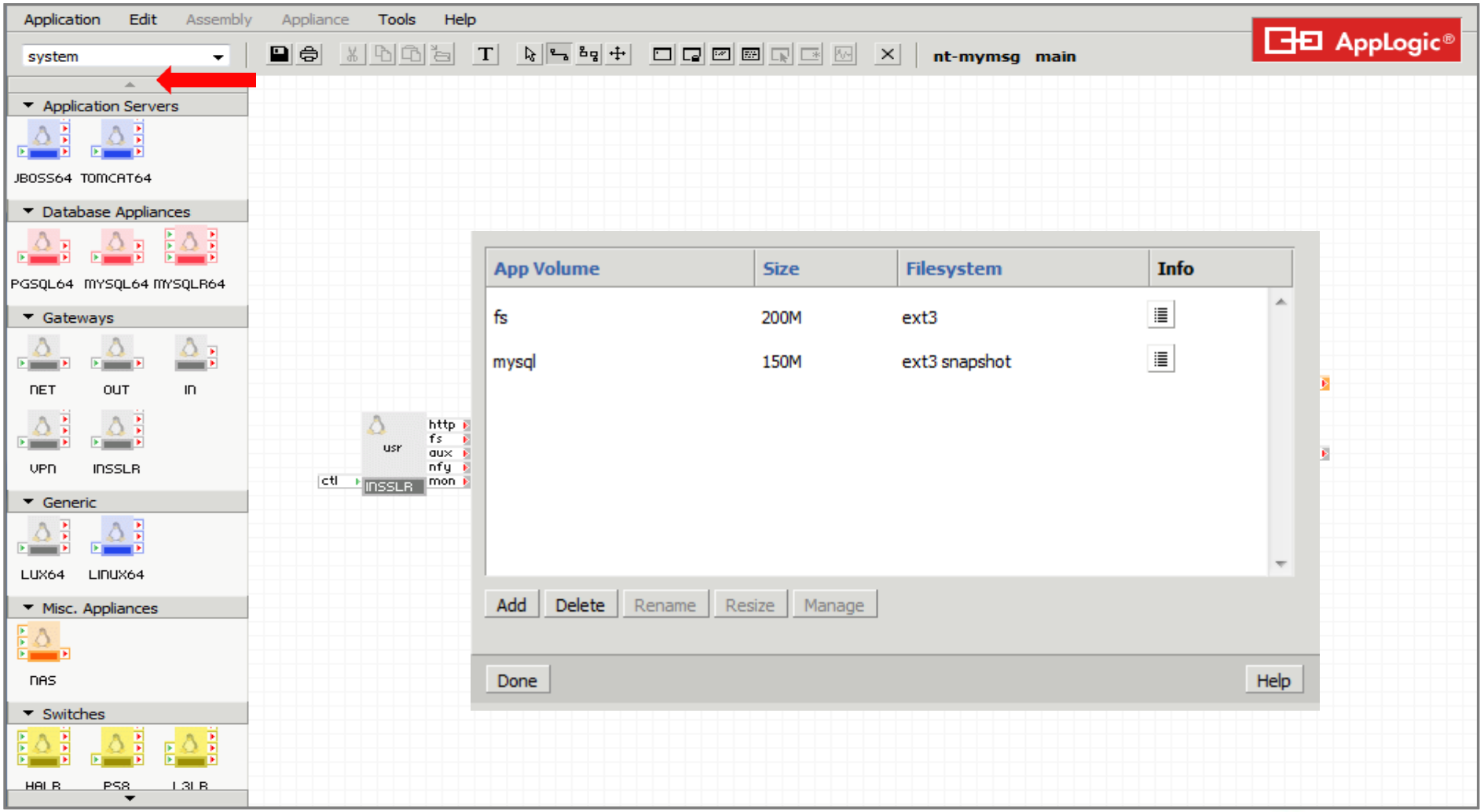
CA AppLogic® for System z
Infrastructure Editor – Assembly



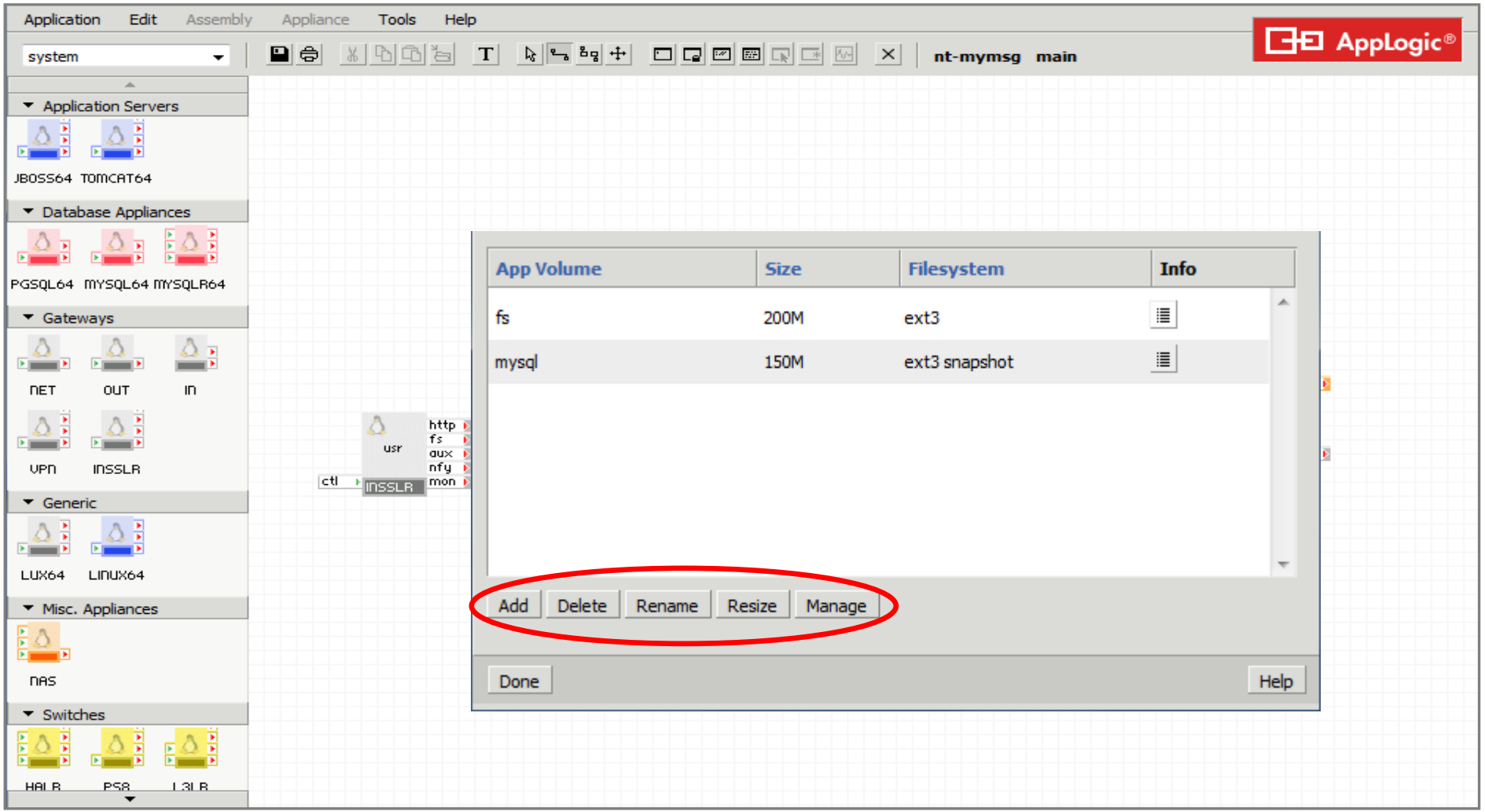
CA AppLogic® for System z
Infrastructure Editor – Assembly



CA AppLogic® for System z
Infrastructure Editor – Assembly

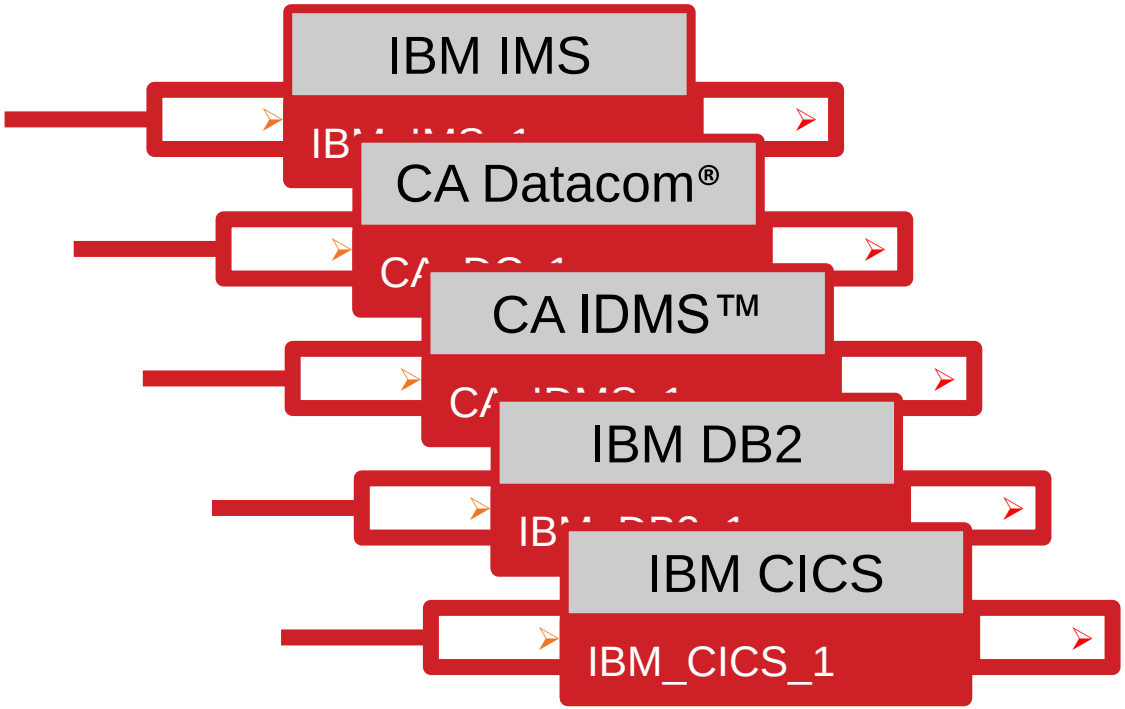


CA AppLogic® for System z
Infrastructure Editor – Assembly



CA AppLogic® for System z
z/OS Service End Point

Five certified z/OS subsystems:



…or create your own

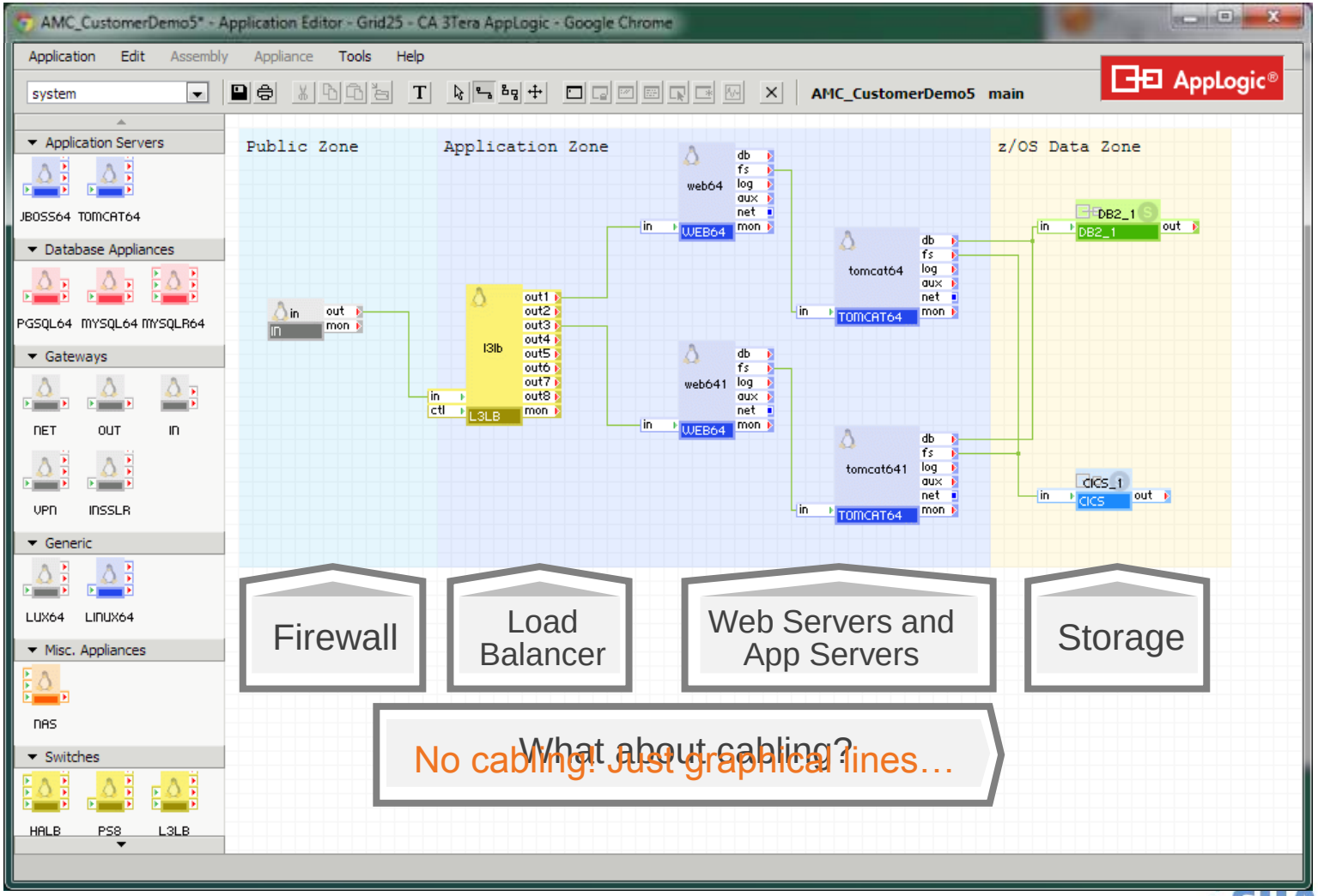
Service Endpoints

- **Behave as connectors to off-grid resources**
- **Simply a package of connection information**
  - IP and port for generic case
  - Available properties customizable per target class known as ‘type’
    - CICS, DB2, myCustomServer,…
  - Instance specific values can be created by z/OS admins and used by AppLogic Application Developers without having *any* z/OS knowledge
  - Connected appliance can retrieve properties for internal use
- **zOS_SvcPt_Admin application delivered for creating SvcPt Types and SvcPts**
- **Allows trivial swap between dev and production databases and many other capabilities**

Agenda

- Current Deployment Options for Linux on System z
- CA AppLogic for System z - The Grid Architecture
- Closer Look at the Solution
- **Using CA AppLogic for System z**
- Summary

Using CA AppLogic® for System z
Putting it all together



Agenda

- Current Deployment Options for Linux on System z
- CA AppLogic for System z - The Grid Architecture
- Closer Look at the Solution
- Using CA AppLogic for System z
- **Summary**

Summary

Linux on System z is the optimal platform for many scenarios

Today's methods are expensive, slow and high risk

CA AppLogic® for System z:

- Simplifies deployment and management of Linux on the mainframe
- Separates the application from the data center infrastructure
- Increases productivity while reducing risk

Interested in Seeing More?

Join us at the CA Technologies Booth in the Share Technology Exchange for a closer look!

Also, visit the CA Linux Management for Mainframe web portal at:

http://www.ca.com/us/mainframe-linux.aspx

AppLogic for Linux on z Documentation

http://doc.3tera.com/AppLogic30z/

Thank You

Contact Information

Summer Spaulding

Sr Principal Engineering Services Architect

CA Technologies

E-mail: summer.spaulding@ca.com

Office: 214-473-1641

Cell: 214-213-9650